Accepted Manuscript

A Discrete Filled Function Algorithm Embedded with Continuous Approximation for Solving Max-Cut Problems

Ai-fan Ling, Cheng-xian Xu, Feng-min Xu

PII: S0377-2217(08)00588-2
DOI: 10.1016/j.ejor.2008.07.026
Reference: EOR 9190

To appear in: *European Journal of Operational Research*

Received Date: 21 July 2007
Revised Date: 22 July 2008
Accepted Date: 22 July 2008

A Discrete Filled Function Algorithm Embedded with Continuous Approximation for Solving Max-Cut Problems *

Ai-fan LING[†], Cheng-xian XU[‡], Feng-min XU[§]

Department of Scientific Computing & Applied Software,

Xi'an Jiaotong University, Xi'an, 710049, PR China

Abstract

In this paper, a discrete filled function algorithm embedded with continuous approximation is proposed to solve max-cut problems. A new discrete filled function is defined for max-cut problems, and properties of the function are studied. In the process of finding an approximation to the global solution of a max-cut problem, a continuation optimization algorithm is employed to find local solutions of a continuous relaxation of the max-cut problem, and then global searches are performed by minimizing the proposed filled function. Unlike general filled function methods, characteristics of max-cut problems are used. The parameters in the proposed filled function need not to be adjusted and are exactly the same for all max-cut problems that greatly increases the efficiency of the filled function method. Numerical results and comparisons on some well known max-cut test problems show that the proposed algorithm is efficient to get approximate global solutions of max-cut problems.

Keywords Combinatorial optimization, Global optimization, Filled function, Max-cut, Continuation method, Local search.

MR(2000)Class

---

*This work is supported by National Natural Science Foundations of China, No. 10671152. The main idea of this article has been presented by Professor Xu at *the* 6^{th} *International Conference on Optimization and Numerical Algebra*, Urumqi, Sept. 2007.

[†]Corresponding author, E-mail: aifanling@yahoo.com.cn

[‡]This author is a member of SKLMSE Laboratory, E-mail: mxxu@mail.xjtu.edu.cn

[§]E-mail: fengminxu@mail.xjtu.edu.cn

1 Introduction

Filled function methods introduced by Ge [8] in 1990 are a class of global optimization methods for general nonlinear continuous optimization problems. Because some good properties of filled function methods, Ge and Huang [9] further extended the filled function method to solve some small scale nonlinear integer programming problems. Recently, several authors proposed some new filled function methods to solve discrete optimization problems with following general form

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ s.t. & \mathbf{x} \in \Omega \subset \mathbb{Z}^n, \end{array} \tag{1.1}$$

where $f : \Omega \to \mathbb{R}$ is a real valued function, and $\mathbb{Z}^n$ is the space of n-dimensional integer column vectors. For instance, Ng et al. [16] dealt with discrete optimization problems using the following filled function,

$$p(\mathbf{x};\mathbf{x}^*;\rho,\mu) = \begin{cases} \mu[f(\mathbf{x}) - f(\mathbf{x}^*)]^2 - \rho \parallel \mathbf{x} - \mathbf{x}^* \|_2^2, & \text{when } f(\mathbf{x}) \geq f(\mathbf{x}^*) \ ; \\ f(\mathbf{x}^*) - f(\mathbf{x}) - \rho\|\mathbf{x} - \mathbf{x}^*\|_2^2, & \text{when } f(\mathbf{x}) \leq f(\mathbf{x}^*), \end{cases} \tag{1.2}$$

where $\mu, \ \rho$ are two adjustable parameters, and $\mathbf{x}^*$ is a local solution of (1.1).

Let $\mathbf{x}_1^*$ be a local minimizer of problem (1.1), the basic idea of the discrete filled function method for (1.1) is to construct an auxiliary function, called filled function, at the point $\mathbf{x}_1^*$, which can be further minimized to get a point, say $\overline{\mathbf{x}}$, in a discrete basin (see Definition 5 below) of $f(\mathbf{x})$ that is lower than the discrete basin containing $\mathbf{x}_1^*$ when $\mathbf{x}_1^*$ is not a global minimizer. Then the minimization of $f(\mathbf{x})$ is restarted from the point $\overline{\mathbf{x}}$ and another local minimizer $\mathbf{x}_2^*$ satisfying $f(\mathbf{x}_2^*) < f(\mathbf{x}_1^*)$ can be obtained. If $\mathbf{x}_2^*$ is still not a global minimizer of $f(\mathbf{x})$ on Ω, then the process is repeated until a global minimizer of $f(\mathbf{x})$ is obtained.

More results and progresses on some new proposed filled functions for discrete optimization problems can be found in [5, 14, 15, 17, 19, 21, 22]. However, at the best of our knowledge, there is very few attempts that have been made for the solution of max-cut problems or other combinatorial optimization problems using the filled function methods.

Max-cut problems are a kind of special discrete optimization problems. Given a graph $G(V;E)$, with node set $V$and edge set E, the problem is to find a partition, $S_1 \subset V$ and $S_2 = V \setminus S_1$, of the set V such that the sum of the weights on the edges connecting the two parts is maximized.

The max-cut problem has long been known to be NP-complete [13], even for any un-weighted graphs [7], and has wide applications in circuit layout design, statistical physics and so on [3]. Approximate algorithms [10, 23], heuristic algorithms [4] and continuous algorithms [18, 20, 21] have been proposed to get approximate solutions of max-cut problems. Based on a semidefinite programming(SDP) relaxation of the max-cut problem, Goemans-Williamson in [10] proposed a 0.878-approximation randomized algorithm for nonnegative weighted graphs. Burer et al. in [4] proposed a rank-2 relaxation and developed a continuous optimization heuristic for solving max-cut problems.

Recently, Xu et al. in [20] (also see [18]) proposed a continuous optimization method to solve max-cut problems, in which a max-cut problem is relaxed into a nonlinear continuous optimization problem with convex constraints. An obvious advantage of the continuous method is that it greatly reduces the CPU-time via without using linear searches and no matrix calculation in each iteration. However, the solution obtained by the continuous method can not be guaranteed as a global minimizer, which motivates us to study global optimization method for the solution of max-cut problems. In this paper, a new filled function is defined and the parameters in the filled function can be exactly estimated for all max-cut problems. Then a discrete filled function algorithm embedded with the continuous approximation is designed for the solution of max-cut problems.

The remainder of the paper is arranged as follows. In section 2, some definitions and preliminaries about discrete filled functions are recalled; In section 3, the continuous algorithm proposed by Xu et al. in [20] is briefly introduced; In section 4, a new discrete filled function for max-cut problems is defined, and properties of the proposed filled function are studied; The parameters of the new filled function are estimated in section 5; The discrete filled function algorithm embedded with the continuous approximation is presented in section 6; Numerical experiments and comparisons on some well known max-cut test problems are reported in section 7.

Throughout the paper, if without special statement, we adopt the following conventions. $\mathbb{R}$, $\mathbb{Z}$, $\mathbb{Z}^n$ and $\mathbb{R}^n$ denote the sets of real numbers, integer numbers, and the spaces of n-dimensional integer column vectors and n-dimensional real column vectors, respectively. $\mathcal{S}^n$, $\mathcal{S}^n_+$ and $\mathcal{S}^n_{++}$ denote the spaces of symmetric matrices, semi-definite positive matrices and positive definite matrices, respectively. $\mathbb{I}$ denotes the n-dimensional unit matrix. Let $\mathbf{x} \in \mathbb{R}^n$, the l_p -norm of $\mathbf{x}$ is denoted by $\|\mathbf{x}\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}}$ $(1 \le p < \infty)$, and set $\mathbb{S} = \{-1, 1\}^n \subset \mathbb{Z}^n$. $\mathbf{e} \in \mathbb{R}^n$ is the column vector with all ones. For any matrix $W \in \mathcal{S}^n$, Diag $(W\mathbf{e})$ is the diagonal matrix with elements of the vector $W\mathbf{e}$ being diagonal entries.

2 Definitions and Preliminaries

In this section, we will give some definitions and lemmas without proofs.

Definition 1 *A sequence* $\{\mathbf{x}^{(i)}\}_{i=-1}^{u}$ *is called a discrete path in the set* $\mathbb{S}$ *between two points* $\mathbf{x}^*$, $\mathbf{x}^{**}$ *in* $\mathbb{S}$, *if* $\mathbf{x}^{(i)} \neq \mathbf{x}^{(j)}$ *for all* $i \neq j$; *and* $\|\mathbf{x}^{(i)} - \mathbf{x}^{(i-1)}\|_p = 2$, *for all* $i = 0, 1, 2, \cdots, u$ *where* $\mathbf{x}^{(-1)} = x^*, \mathbf{x}^{(u)} = \mathbf{x}^{**}$. *In addition, if either* $\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_p < \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_p$, *or* $\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_p > \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_p$ *holds for all* i, *then the sequence is called a strict discrete path in* $\mathbb{S}$. *If a (strict) discrete path exists for two given points* $\mathbf{x}^* \in \mathbb{S}$, $\mathbf{x}^{**} \in \mathbb{S}$, $\mathbf{x}^* \neq \mathbf{x}^{**}$, *then* $\mathbf{x}^*$ *and* $\mathbf{x}^{**}$ *are said to be (strictly) path-wise connected in* $\mathbb{S}$.

For simplicity, we call a (strict) discrete path as a (strict) path. From the definition of the path, it is clear that any two distinct points $\mathbf{x}, \mathbf{y} \in \mathbb{S}$ are (strictly) path-wise connected, and hence $\mathbb{S}$ is a (strictly) path-wise connected domain.

Definition 2 *For any* $\mathbf{x} \in \mathbb{S}$, *and any positive integer* $K, 1 \leq K \leq n$, *the* K*-neighborhood of the point* $\mathbf{x}$ *under the* l_p*-norm is defined by*

$$N(\mathbf{x}, K) = \{\mathbf{y} \in \mathbb{S} \ : \ \|\mathbf{y} - \mathbf{x}\|_p \leq 2 \cdot K^{\frac{1}{p}}\}.$$

Particular, if $K = 1$, we write the 1-neighborhood $N(\mathbf{x}, 1)$ of $\mathbf{x}$ as $N(\mathbf{x})$. The boundary of a K-neighborhood $N(\mathbf{x}, K)$ is defined as

$$\partial N(\mathbf{x}, K) = \{\mathbf{y} \in \mathbb{S} \ : \ \|\mathbf{y} - \mathbf{x}\|_p = 2 \cdot K^{\frac{1}{p}}\}.$$

It can be verified that $|\partial N(\mathbf{x}, K)| = \binom{n}{K}$. Let $\overline{\mathbf{x}}$ be a point in $\partial N(\mathbf{x}, K)$, then $\overline{\mathbf{x}}$ differs from the point $\mathbf{x}$ in only K elements. The following Lemma is obvious from Definition 2.

Lemma 2.1 *For every* $K \in \mathbb{Z}, 1 \leq K \leq n$, *and any* $\mathbf{x} \in \mathbb{S}$, *the number of points in* $N(\mathbf{x}, K)$ *is* $1 + \sum_{i=1}^{K} \binom{n}{i}$. *Especially, when* $K = 1$, $|N(\mathbf{x})| = n + 1$. □

Definition 3 *A point* $\mathbf{x}^* \in \mathbb{S}$ *is called a local discrete minimizer of the function* f *over* $\mathbb{S}$, *if* $f(\mathbf{x}^*) \leq f(\mathbf{x})$ *for all* $\mathbf{x} \in N(\mathbf{x}^*)$. *Furthermore, if* $f(\mathbf{x}^*) \leq f(\mathbf{x})$ *for all* $\mathbf{x} \in \mathbb{S}$, *then* $\mathbf{x}^*$ *is called a global discrete minimizer of* f *over* $\mathbb{S}$.

Let $\mathbf{e}_i \in \mathbb{R}^n$ be the i^{th} unit directional vector, and $\mathcal{D} = \{\mathbf{d} : \mathbf{d} = \pm 2 \cdot \mathbf{e}_i, \ i = 1, \cdots, n\}$. Then for any $\mathbf{x} \in \mathbb{S}$, there exists a direction $\mathbf{d} \in \mathcal{D}$ such that $\mathbf{x} + \mathbf{d} \in N(\mathbf{x}) \subset \mathbb{S}$.

Definition 4 *For any $\mathbf{x} \in \mathbb{S}$, a direction $\mathbf{d} \in \mathcal{D}$ is called a descent direction of the function f at $\mathbf{x}$ over $\mathbb{S}$, if $\mathbf{x} + \mathbf{d} \in \mathbb{S}$ and $f(\mathbf{x} + \mathbf{d}) < f(\mathbf{x})$. In addition, $\mathbf{d}^*$ is called the steepest discrete descent direction of the function f at $\mathbf{x}$ over $\mathbb{S}$, if $f(\mathbf{x} + \mathbf{d}^*) \leq f(\mathbf{x} + \mathbf{d})$ holds for all the descent directions $\mathbf{d}$ of the function f at $\mathbf{x}$ over $\mathbb{S}$.*

Definition 5 *Let $\mathbf{x}^*$ be a local minimizer of the function f over $\mathbb{S}$, B^* is said to be a discrete basin of the function f at $\mathbf{x}^*$, if $B^* \subset \mathbb{S}$ is a path-wise connected domain which contains $\mathbf{x}^*$, the steepest discrete descent trajectory from any point in B^* converges to $\mathbf{x}^*$, and the steepest discrete descent trajectory from any point in $\mathbb{S} \backslash B^*$ does not converge to $\mathbf{x}^*$.*

Note that Definition 1, 3, 4 and 5 for max-cut problems above are modifications of the corresponding to definitions in [16].

Definition 6 *[19] Let $\mathbf{x}^*$ and $\mathbf{x}^{**}$ be two distinct local minimizers of the function f over $\mathbb{S}$, the discrete basin B^{**} of f at $\mathbf{x}^{**}$ is said to be lower (or higher) than the discrete basin B^* of f at $\mathbf{x}^*$, if $f(\mathbf{x}^{**}) < f(\mathbf{x}^*)$ (or $f(\mathbf{x}^{**}) > f(\mathbf{x}^*)$).*

Definition 7 *[16] Let $\mathbf{x}^*$ be a discrete minimizer of the function f over $\mathbb{S}$, and B^* be the discrete basin of f at $\mathbf{x}^*$. A function $H : \mathbb{S} \to \mathbb{R}$ is said to be a discrete filled function of f at $\mathbf{x}^*$, if it satisfies the following conditions:*

1. *$\mathbf{x}^*$ is a strict discrete local maximizer of H over $\mathbb{S}$.*
2. *H has no discrete local minimizers in B^* and in any discrete basin of f higher than B^*.*
3. *if f has a discrete basin , B^{**} say, lower than B^*, then there is a point $\overline{\mathbf{x}} \in B^{**}$ that minimizes H on the path $\{\mathbf{x}^*, \cdots, \overline{\mathbf{x}}, \cdots, \mathbf{x}^{**}\}$ in $\mathbb{S}$, where $\mathbf{x}^{**}$ is a minimizer of f in B^{**}.*

3 The Continuation Method for Max-Cut Problems

Let $W = (w_{ij})_{n \times n}$ be the symmetric weighted adjacency matrix for a given graph $G(V, E)$ with $w_{ij} \neq 0$ for $(i,j) \in E$ and $w_{ij} = 0$ for $(i,j) \notin E$. The max-cut problem can be expressed as the following discrete quadratic optimization problem:

$$\text{(MC)}: \qquad mc^* = \max \quad \mathbf{x}^T \widehat{L} \mathbf{x} \qquad s.t. \quad \mathbf{x} \in \mathbb{S}$$

where $\widehat{L}$ =Diag($W\mathbf{e}$) $- W \in \mathcal{S}_+^n$ is the Laplace matrix of the graph. Since $\widehat{L} + \sigma\mathbb{I} \in \mathcal{S}_{++}^n$ and $\mathbf{x}^T(\widehat{L} + \sigma\mathbb{I})\mathbf{x} = \mathbf{x}^T\widehat{L}\mathbf{x} + \sigma\mathbf{x}^T\mathbf{x} = \mathbf{x}^T\widehat{L}\mathbf{x} + n\sigma$ for any $\sigma > 0$ and for any $\mathbf{x} \in \mathbb{S}$, without loss of generality, we always assume that the matrix $\widehat{L} \in \mathcal{S}_{++}^n$ with $\widehat{l}_{ii} > 0$ for all $i = 1, \cdots, n$. Let $L = -\widehat{L}$, then the problem (MC) can be written as

$$\text{(MMC)}: \qquad \begin{array}{ll} \min & f(\mathbf{x}) = \mathbf{x}^T L\mathbf{x} \\ s.t. & \mathbf{x} \in \mathbb{S} \end{array}$$

where L is a negative definite matrix with diagonal entries $l_{ii} < 0$ for all $i = 1, \cdots, n$.

Xu et al. in [20] proposed a continuous relaxation method to solve problem (MMC), in which problem (MMC) was relaxed into the following nonlinear optimization problem:

$$\text{(MX)}. \quad \begin{array}{ll} \min & f(\mathbf{x}) = \mathbf{x}^T L\mathbf{x} \\ s.t. & B(x_i, 1) \le 0, i = 1, \cdots, n, \\ & \|\mathbf{x}\|_2^2 \le n \end{array}$$

where

$$B(x_i, 1) = \begin{cases} \frac{1+x_i}{2}\log(\frac{1+x_i}{2}) + \frac{1-x_i}{2}\log(\frac{1-x_i}{2}), & x_i \in (-1, 1); \\ 0, & x_i = -1, \text{ or } x_i = 1. \end{cases}$$

is strictly convex with respect to $x_i \in (-1, 1)$ (see [20]). Let $\mathbf{x}^*$ and $\mathbf{x}^{**}$ be global minimizers of problems (MMC) and (MX), respectively, then $f(\mathbf{x}^*) \ge f(\mathbf{x}^{**})$ or $mc^* \le -f(\mathbf{x}^{**})$, and $\overline{\mathbf{x}}$ =sign($\mathbf{x}^{**}$) $\in \mathbb{S}$ is accepted as an approximate solution of (MC), where sign($\cdot$) is the sign function. Furthermore, improvements on $\overline{\mathbf{x}}$ can be made by using local searches from the point $\overline{\mathbf{x}}$.

Assume that $\mathbf{x}_k$ is a feasible solution of (MX), and $\mathbf{g}_k = 2L\mathbf{x}_k$ is the gradient of $f(\mathbf{x})$ at $\mathbf{x}_k$. The continuous algorithm generates a new point by $\mathbf{x}_{k+1} = \mathbf{g}_k\|\mathbf{x}_k\|_2/\|\mathbf{g}_k\|_2$. Define $\mathbf{d}_k = \mathbf{x}_{k+1} - \mathbf{x}_k = \mathbf{g}_k\|\mathbf{x}_k\|_2/\|\mathbf{g}_k\|_2 - \mathbf{x}_k$ as the search direction at $\mathbf{x}_k$. Then $\mathbf{d}_k$ and the continuous algorithm have the following properties (see [20]).

1. $\mathbf{x}_{k+1}$ is feasible to problem (MX).
2. Let $\mathbf{d}_k \neq 0$, then $\mathbf{d}_k$ is a feasible descent direction of problem (MX) at $\mathbf{x}_k$.
3. If $\mathbf{d}_k = 0$, then $\mathbf{x}_k$ is a KKT-point of problem (MX).
4. If $\mathbf{d}_k \neq 0$, for all $k > 0$, then $\|\mathbf{d}_k\|_2 \to 0$.

These properties indicate that the continuous algorithm either terminates at a KKT point in a finite steps or converges to a KKT point. The characteristics of the continuous algorithm are that the dimension of the relaxed continuous optimization problem is not increased, no line searches and no matrix calculations are required in the implementation of the algorithm. These properties greatly reduce the CPU-time of the algorithm and save the memory. For simplicity, we denote the continuation algorithm by Algorithm (CA) in the rest of this paper, and if without special statement, the function f is only viewed as the objective function in problem (MMC).

Since the function $f(\mathbf{x})$ is concave, the solution obtained by Algorithm (CA) can not be guaranteed to be a global optimal solution. Therefore, a discrete filled function algorithm is proposed here to find a global minimizer of problem (MMC), in which the algorithm (CA) is used to find local minimizers of problem (MMC).

4 A New Discrete Filled Function and Its Properties

In this section, we propose a new discrete filled function that can exploit the special structure of max-cut problems. The filled function is defined as follows.

$$H_p(\mathbf{x};\mathbf{x}^*;\alpha,\beta) = \begin{cases} \frac{-1}{a+f(\mathbf{x})-f(\mathbf{x}^*)} - \beta\|\mathbf{x}-\mathbf{x}^*\|_p, & f(\mathbf{x}) \geq f(\mathbf{x}^*); \\ \frac{-\alpha}{a+f(\mathbf{x}^*)-f(\mathbf{x})} - \beta\|\mathbf{x}-\mathbf{x}^*\|_p + \frac{\alpha-1}{a}, & f(\mathbf{x}) \leq f(\mathbf{x}^*). \end{cases} \tag{4.1}$$

where $\mathbf{x}^*$ is a local minimizer of the function f, $a > 0$ is a constant, and $\alpha > 0, \beta > 0$ are two parameters. The term $\frac{\alpha-1}{a}$ ensures that the function $H_p(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ is continuous in $\mathbb{R}^n$. For simplicity, in the rest of the paper, we only consider the case of $p = 1$, and denote $H_1(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ by $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$, but it is not hard to extend all results of this paper to the case of $1 < p < \infty$. Two obvious advantages of the filled function $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ can be obtained from the analysis below.

- In order to minimize the filled function H from a neighbor point $\overline{\mathbf{x}}$ of $\mathbf{x}^*$, we only need to minimize the function f from $\overline{\mathbf{x}}$ along a direction that is away from the minimizer $\mathbf{x}^*$ (see subsection 6.4), instead of finding a descent direction of the filled function $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ at $\overline{\mathbf{x}}$.
- The values of the parameters α and β in the filled function H can be exactly confirmed for all max-cut problems (see section 5). This is significant for the solution of large scale max-cut problems.

Now we will show that the function $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ is a desired discrete filled function of f at point $\mathbf{x}^*$. The following lemma plays an important role in the analysis of this section.

Lemma 4.1 *Let matrix L be given and $\mathbf{x} \in \mathbb{S}$, then for any $\mathbf{y} \in N(\mathbf{x})$, we have*

$$|f(\mathbf{y}) - f(\mathbf{x})| \leq 4qM + l_0 \ ,$$

where $q = \max\{q_i : i = 1, \cdots, n\} \leq n$ and $M = \max\{\|\mathbf{l}_i\|_1 : i = 1, \cdots, n\}$, $\mathbf{l}_i$ is the i^{th} column of the matrix L, q_i $(i = 1, \cdots, n)$ is the number of nonzero elements in the vector $\mathbf{l}_i$ $(i = 1, \cdots, n)$, and $l_0 = \max\{-4l_{ii} : i = 1, \cdots, n\} > 0$.

Proof. Let $\mathbf{x}^m \in N(\mathbf{x})$ $(m = 1, 2, \cdots, n)$, where the superscript m denotes that only the m^{th} element of $\mathbf{x}^m$ differs from the m^{th} element of $\mathbf{x}$, i.e.

$$x_i^m = \begin{cases} x_i, & i = 1, 2, \cdots, n, i \neq m; \\ -x_i, & i = m. \end{cases}$$

Since $l_{mm} < 0, m = 1, 2, \cdots, n$, we have

$$\begin{aligned} |f(\mathbf{x}^m) - f(\mathbf{x})| &= |(\mathbf{x}^m)^T L \mathbf{x}^m - \mathbf{x}^T L \mathbf{x}| \\ &= |\sum_{i \neq m}^{n} \sum_{j \neq m}^{n} x_i l_{ij} x_j + x_m l_{mm} x_m - 2 \sum_{i \neq m}^{n} x_i l_{im} x_m - \sum_{i,j}^{n} x_i l_{ij} x_j| \\ &= 4|\sum_{i=1, i \neq m}^{n} x_i l_{im} x_m| \\ &= 4|x_m \cdot \mathbf{x}^T \mathbf{l}_m - x_m l_{mm} x_m| \\ &\leq 4|\mathbf{x}^T \mathbf{l}_m| - 4l_{mm} \leq 4q_m \|\mathbf{l}_m\|_1 - 4l_{mm} \\ &\leq 4qM + l_0, \ m = 1, 2, \cdots, n. \end{aligned}$$

This completes the proof. □

Lemma 4.2 *Let $\mathbf{x} \in \mathbb{S}$, then there exist an integer $K (1 \leq K \leq n)$ such that*

$$|f(\mathbf{y}) - f(\mathbf{x})| \leq K(4qM + l_0)$$

holds for all $\mathbf{y} \in \mathbb{S} \setminus \{\mathbf{x}\}$, where constants q, M, and l_0 are given in Lemma 4.1.

Proof. Let $\mathbf{y} \in \mathbb{S}$, $\mathbf{y} \neq \mathbf{x}$, then there exists an integer $K (1 \leq K \leq n)$ such that $\mathbf{y} \in \partial N(\mathbf{x}, K)$. Hence, there must exist at least one path $\{\mathbf{x}_j\}_{j=0}^{K}$ from $\mathbf{x}$ to $\mathbf{y}$ satisfying $\mathbf{x}_0 = \mathbf{x}$, $\mathbf{x}_K = \mathbf{y}$ and $\mathbf{x}_j \in N(\mathbf{x}_{j-1}), j = 1, \cdots, K$. Since

$$\begin{aligned} |f(\mathbf{y}) - f(\mathbf{x})| &= |f(\mathbf{x}_K) - f(\mathbf{x}_{K-1}) + f(\mathbf{x}_{K-1}) - f(\mathbf{x}_{K-2}) + \cdots f(\mathbf{x}_1) - f(\mathbf{x}_0)| \\ &\leq \sum_{j=1}^{K} |f(\mathbf{x}_j) - f(\mathbf{x}_{j-1})|, \end{aligned}$$

it then follows from Lemma 4.1 that

$$|f(\mathbf{y}) - f(\mathbf{x})| \leq \sum_{j=1}^{K} (4qM + l_0) = K(4qM + l_0). \quad \square$$

Theorem 4.3 *Let $\mathbf{x}^*$ be a local minimizer of the function f on $\mathbb{S}$ and $\alpha > 0$. If*

$$\beta > \frac{4qM + l_0}{2a(a + 4qM + l_0)}, \tag{4.2}$$

then the function $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$ satisfies the first condition of Definition 7, i.e. $\mathbf{x}^*$ is a strict local maximizer of $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$ on $\mathbb{S}$. Furthermore, if $\mathbf{x}^*$ is a global minimizer of f on $\mathbb{S}$, then $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta) < -1/a$ for all $\mathbf{x} \in \mathbb{S} \setminus \{\mathbf{x}^*\}$, that is, $\mathbf{x}^*$ is a global maximizer of $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$ on $\mathbb{S}$.

Proof. Let $\mathbf{x}^*$ be a local minimizer of f on $\mathbb{S}$. Since $f(\mathbf{x}) \geq f(\mathbf{x}^*)$ and $\|\mathbf{x} - \mathbf{x}^*\|_1 = 2$ for any $\mathbf{x} \in N(\mathbf{x}^*) \setminus \{\mathbf{x}^*\}$. It follows from Lemma 4.1 that

$$\begin{aligned} H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta) &= \tfrac{-1}{a + f(\mathbf{x}) - f(\mathbf{x}^*)} - \beta \|\mathbf{x} - \mathbf{x}^*\|_1 \\ &\leq \tfrac{-1}{a + 4qM + l_0} - 2\beta. \end{aligned} \tag{4.3}$$

Since β satisfies (4.2), the right hand side of (4.3) is strictly less than $-1/a$, that is

$$H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta) < -\frac{1}{a} = H(\mathbf{x}^*; \mathbf{x}^*; \alpha, \beta)$$

Hence, $\mathbf{x}^*$ is a strict local maximizer of $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$ on $\mathbb{S}$.

If $\mathbf{x}^*$ is a global minimizer of f on $\mathbb{S}$, then for any given $\mathbf{x} \in \mathbb{S} \setminus \{\mathbf{x}^*\}$, there exists an integer $K (1 \leq K \leq n)$ depending on the $\mathbf{x}$, such that $\mathbf{x} \in \partial N(\mathbf{x}^*, K)$. Hence, from Lemma 4.2,

$$\begin{aligned} H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta) &= \tfrac{-1}{a + f(\mathbf{x}) - f(\mathbf{x}^*)} - \beta \|\mathbf{x} - \mathbf{x}^*\|_1 \\ &\leq \tfrac{-1}{a + K(4qM + l_0)} - 2\beta K. \end{aligned} \tag{4.4}$$

Since

$$\beta > \frac{4qM + l_0}{2a(a + 4qM + l_0)} \geq \frac{(4qM + l_0)}{2a[a + K(4qM + l_0)]}.$$

Hence the right hand side of (4.4) satisfies

$$\begin{aligned} \tfrac{-1}{a + K(4qM + l_0)} - 2\beta K &< \tfrac{-1}{a + K(4qM + l_0)} - \tfrac{K(4qM + l_0)}{a[a + K(4qM + l_0)]} \\ &= \tfrac{-1}{a + K(4qM + l_0)} - (\tfrac{1}{a} - \tfrac{1}{a + K(4qM + l_0)}) \\ &= -\tfrac{1}{a} = H(\mathbf{x}^*; \mathbf{x}^*; \alpha, \beta). \end{aligned}$$

This completes the proof. □

Lemma 4.4 *Let $\mathbf{x}^*$ be a local minimizer of the function f on $\mathbb{S}$, $\overline{\mathbf{x}} \in \mathbb{S}$, and $f(\overline{\mathbf{x}}) \geq f(\mathbf{x}^*)$. If there exists $\overline{\mathbf{d}} \in \mathcal{D}$, such that $\overline{\mathbf{x}} + \overline{\mathbf{d}} \in \mathbb{S}$, $f(\overline{\mathbf{x}} + \overline{\mathbf{d}}) \geq f(\mathbf{x}^*)$ and $\|\overline{\mathbf{x}} + \overline{\mathbf{d}} - \mathbf{x}^*\|_1 > \|\overline{\mathbf{x}} - \mathbf{x}^*\|_1$, and if β satisfies (4.2), then we have*

$$H(\overline{\mathbf{x}} + \overline{\mathbf{d}}; \mathbf{x}^*; \alpha, \beta) < H(\overline{\mathbf{x}}; \mathbf{x}^*; \alpha, \beta) < H(\mathbf{x}^*; \mathbf{x}^*; \alpha, \beta) = -\frac{1}{a}, \tag{4.5}$$

Proof. Since $\|\overline{\mathbf{x}}+\overline{\mathbf{d}}-\mathbf{x}^*\|_1 > \|\overline{\mathbf{x}}-\mathbf{x}^*\|_1$, $f(\overline{\mathbf{x}}) \geq f(\mathbf{x}^*)$, and $f(\overline{\mathbf{x}}+\overline{\mathbf{d}}) \geq f(\mathbf{x}^*)$. It follows that $\|\overline{\mathbf{x}}+\overline{\mathbf{d}}-\mathbf{x}^*\|_1 - \|\overline{\mathbf{x}}-\mathbf{x}^*\|_1 = 2$, and

$$\begin{aligned} & H(\overline{\mathbf{x}}+\overline{\mathbf{d}};\mathbf{x}^*;\alpha,\beta) - H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta) \\ &= \tfrac{-1}{a+f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\mathbf{x}^*)} + \tfrac{1}{a+f(\overline{\mathbf{x}})-f(\mathbf{x}^*)} - 2\beta \\ &= \tfrac{f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\overline{\mathbf{x}})}{[a+f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\mathbf{x}^*)][a+f(\overline{\mathbf{x}})-f(\mathbf{x}^*)]} - 2\beta. \end{aligned} \tag{4.6}$$

If $f(\overline{\mathbf{x}}+\overline{\mathbf{d}}) \leq f(\overline{\mathbf{x}})$, then $H(\overline{\mathbf{x}}+\overline{\mathbf{d}};\mathbf{x}^*;\alpha,\beta) < H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta)$ for any $\beta > 0$. If $f(\overline{\mathbf{x}}+\overline{\mathbf{d}}) \geq f(\overline{\mathbf{x}})$, since $f(\overline{\mathbf{x}}) \geq f(\mathbf{x}^*)$, we have

$$\begin{aligned} & [a+f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\mathbf{x}^*)][a+f(\overline{\mathbf{x}})-f(\mathbf{x}^*)] \\ &\geq a^2 + a[f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\mathbf{x}^*)+f(\overline{\mathbf{x}})-f(\mathbf{x}^*)] \\ &= a^2 + a[f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\overline{\mathbf{x}})+2(f(\overline{\mathbf{x}})-f(\mathbf{x}^*))] \\ &\geq a^2 + a[f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\overline{\mathbf{x}})] > 0. \end{aligned}$$

Hence

$$H(\overline{\mathbf{x}}+\overline{\mathbf{d}};\mathbf{x}^*;\alpha,\beta) - H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta) \leq \frac{f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\overline{\mathbf{x}})}{a^2+a[f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\overline{\mathbf{x}})]} - 2\beta.$$

Let $g(t) = \frac{t}{a^2+at}, t \geq 0$, then it can be verified that $g(t)$ is a monotonically increasing function with respect to t in the interval $[0,\infty)$. Thus from Lemma 4.1

$$\begin{aligned} H(\overline{\mathbf{x}}+\overline{\mathbf{d}};\mathbf{x}^*;\alpha,\beta) - H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta) &\leq g(f(\overline{\mathbf{x}}+\overline{\mathbf{d}})-f(\overline{\mathbf{x}})) - 2\beta \\ &\leq g(4qM+l_0) - 2\beta \\ &= \tfrac{4qM+l_0}{a^2+a(4qM+l_0)} - 2\beta. \end{aligned}$$

When β satisfies (4.2),

$$H(\overline{\mathbf{x}}+\overline{\mathbf{d}};\mathbf{x}^*;\alpha,\beta) < H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta)$$

holds. The first inequality in (4.5) is proved. The second inequality in (4.5) directly follows from Theorem 4.3. □

For any given point $\overline{\mathbf{x}} \in \mathbb{S}$ with $f(\overline{\mathbf{x}}) \geq f(\mathbf{x}^*)$, let

$$\mathcal{D}_{\overline{\mathbf{x}}} = \{\overline{\mathbf{d}} \in \mathcal{D} : \overline{\mathbf{x}}+\overline{\mathbf{d}} \in \mathbb{S}, f(\overline{\mathbf{x}}+\overline{\mathbf{d}}) \geq f(\mathbf{x}^*), \|\overline{\mathbf{x}}+\overline{\mathbf{d}}-\mathbf{x}^*\|_1 > \|\overline{\mathbf{x}}-\mathbf{x}^*\|_1\}.$$

Lemma 4.4 indicates that if $\mathcal{D}_{\overline{\mathbf{x}}}$ is nonempty, then $\overline{\mathbf{x}}$ is not a local minimizer of the function H when β satisfies (4.2). Especially, we have the following result.

Theorem 4.5 *Let $\mathbf{x}^*$ be a local minimizer of the function f on $\mathbb{S}$, and B^* be the discrete basin of f at $\mathbf{x}^*$. If β satisfies (4.2), then $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$ has no any local minimizer in B^* and in any basin higher than B^*.*

Proof. Let U^* denote the union of all basins of f higher than B^*, it is clear from definition 6 that $f(\mathbf{x}) > f(\mathbf{x}^*)$ holds for any point $\mathbf{x} \in U^*$. Note that $-\mathbf{x}^* \notin U^*$ via $f(-\mathbf{x}^*) = f(\mathbf{x}^*)$ and $-\mathbf{x}^* \notin B^*$ via the fact that $-\mathbf{x}^*$ oneself is a minimizer of f. Hence for any $\overline{\mathbf{x}} \in B^*$ or $\overline{\mathbf{x}} \in U^*$, it follows that $\overline{\mathbf{x}} \neq -\mathbf{x}^*$. Thus, there exists at least a direction $\overline{\mathbf{d}} \in \mathcal{D}$ at $\overline{\mathbf{x}}$, such that $\overline{\mathbf{x}}+\overline{\mathbf{d}} \in \mathbb{S}$ and $\|\overline{\mathbf{x}}+\overline{\mathbf{d}}-\mathbf{x}^*\|_1 > \|\overline{\mathbf{x}}-\mathbf{x}^*\|_1$. If $f(\overline{\mathbf{x}}+\overline{\mathbf{d}}) < f(\mathbf{x}^*)$, then we can obtain a point, $\mathbf{x}^{**}$ say, satisfying $f(\mathbf{x}^{**}) < f(\mathbf{x}^*)$ by minimizing the function f from $\overline{\mathbf{x}}$. It means $\overline{\mathbf{x}}$ is in a basin of f lower than the basin B^*. This contradicts with $\overline{\mathbf{x}} \in B^*$ or $\overline{\mathbf{x}} \in U^*$. Thus we have $f(\overline{\mathbf{x}} + \overline{\mathbf{d}}) \geq f(\mathbf{x}^*)$ and the set $\mathcal{D}_{\overline{\mathbf{x}}}$ is nonempty. It then follows from Lemma 4.4 that $\overline{\mathbf{x}}$ is not a local minimizer of the filled function $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$.□

From Lemma 4.4 and Theorem 4.5, we obtain that when β satisfies (4.2), for any $\mathbf{x} \in U^*$ or $\mathbf{x} \in B^*, \mathbf{x} \neq \mathbf{x}^*$, if $\mathbf{d}$ satisfies $\|\mathbf{x}+\mathbf{d}-\mathbf{x}^*\|_1 > \|\mathbf{x}-\mathbf{x}^*\|_1$ and $f(\mathbf{x}+\mathbf{d}) \geq f(\mathbf{x}^*)$, then $\mathbf{d}$ is a descent direction of H at $\mathbf{x}$ and $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$ satisfies the second condition of definition 7. In addition, Lemma 4.4 indicates that when $\overline{\mathbf{x}}$ lies in a basin of the function f lower than the current basin B^*, if the set $\mathcal{D}_{\overline{\mathbf{x}}}$ is nonempty, then $\overline{\mathbf{x}}$ can not be a minimizer of the function $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$, which arises a question whether the function $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$ has local minimizer on $\mathbb{S}$. The following result gives the answer.

Theorem 4.6 *Let $\mathbf{x}^*$, $\mathbf{x}^{**}$ be two distinct minimizers of function f satisfying $f(\mathbf{x}^*) > f(\mathbf{x}^{**})$, and B^*, B^{**} be two neighboring basins of the function f at $\mathbf{x}^*$, $\mathbf{x}^{**}$ in $\mathbb{S}$, respectively. Then the following conclusions hold.*

1. *There exists a point $\overline{\mathbf{x}} \in B^{**}$ and a descent direction $\overline{\mathbf{d}}$ of f at $\overline{\mathbf{x}}$ satisfying*

$$f(\overline{\mathbf{x}} + \overline{\mathbf{d}}) < f(\mathbf{x}^*) \leq f(\overline{\mathbf{x}}). \tag{4.7}$$

   and a strict path $\{\mathbf{x}^{(i)}\}_{i=0}^{u}$ in $\mathbb{S}$ between $\mathbf{x}^*(= \mathbf{x}^{(0)})$ and $\overline{\mathbf{x}}(= \mathbf{x}^{(u)})$, such that

$$f(\mathbf{x}^{(i)}) \geq f(\mathbf{x}^*) \tag{4.8}$$

   holds for all $i = 1, \cdots, u$.

2. *Furthermore, if β satisfies (4.2) and*

$$\alpha > \frac{2a\beta M_1 M_2 + M_1[f(\overline{\mathbf{x}}) - f(\mathbf{x}^*)]}{M_2[f(\mathbf{x}^*) - f(\overline{\mathbf{x}} + \overline{\mathbf{d}})]}, \tag{4.9}$$

then $\overline{\mathbf{x}}$ *is a minimizer of* $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ *on the path* $\{\mathbf{x}^{(i)}\}_{i=0}^{u+v}$ *in* $\mathbb{S}$*, where* $\mathbf{x}^{(u+1)} = \overline{\mathbf{x}}+\overline{\mathbf{d}}$*,* $\mathbf{x}^{(u+v)} = \mathbf{x}^{**}$*, and*

$$M_1 = a + f(\mathbf{x}^*) - f(\overline{\mathbf{x}}+\overline{\mathbf{d}}), M_2 = a + f(\overline{\mathbf{x}}) - f(\mathbf{x}^*). \tag{4.10}$$

Proof. Let $\mathbf{x}^{**} \in \partial N(\mathbf{x}^*, K)$ for some integer $K(1 < K < n)$, then there exists a strict path $\{\mathbf{x}^{(i)}\}_{i=0}^{K}$ with $\mathbf{x}^{(0)} = \mathbf{x}^*, \mathbf{x}^{(K)} = \mathbf{x}^{**}$ satisfying $\|\mathbf{x}^{(i)} - \mathbf{x}^*\|_1 < \|\mathbf{x}^{(i+1)} - \mathbf{x}^*\|_1, i = 0, 1, \cdots, K-1$. Since $f(\mathbf{x}^{**}) < f(\mathbf{x}^*)$, there exists a point, $\mathbf{x}^{(u)}(1 < u < K)$ say, on the path $\{\mathbf{x}^{(i)}\}_{i=0}^{K}$ such that

$$f(\mathbf{x}^{(u)}) \geq f(\mathbf{x}^*), \qquad f(\mathbf{x}^{(u+1)}) < f(\mathbf{x}^*).$$

Hence, if we take $\overline{\mathbf{x}} = \mathbf{x}^{(u)}$ and $\overline{\mathbf{d}} = \mathbf{x}^{(u+1)} - \mathbf{x}^{(u)}$, then point $\overline{\mathbf{x}}$ and the direction $\overline{\mathbf{d}}$ satisfy (4.7). The strict path $\{\mathbf{x}^{(i)}\}_{i=0}^{u}$ with $\mathbf{x}^* = \mathbf{x}^{(0)}$ and $\overline{\mathbf{x}} = \mathbf{x}^{(u)}$ satisfies (4.8). This proves the first conclusion.

To prove the second conclusion, it is sufficient to show $H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta) < H(\mathbf{x}^{(u+1)};\mathbf{x}^*;\alpha,\beta)$, and $H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta) < H(\mathbf{x}^{(u-1)};\mathbf{x}^*;\alpha,\beta)$ hold on the path. Since $\mathbf{x}^*$ is a local minimizer of f on $\mathbb{S}$ and $\mathbf{x}^{(1)} \in B^*$, it follows from Theorem 4.3 that $H(\mathbf{x}^{(1)};\mathbf{x}^*;\alpha,\beta) < H(\mathbf{x}^*;\mathbf{x}^*;\alpha,\beta) = -1/a$ holds when β satisfies (4.2).

Since $\{\mathbf{x}^{(i)}\}_{i=0}^{u}$ is a strict path starting from $\mathbf{x}^*$, and $\|\mathbf{x}^{(i-1)} - \mathbf{x}^*\|_1 < \|\mathbf{x}^{(i)} - \mathbf{x}^*\|_1$ holds for all $i = 1, \cdots, u$. In view of (4.8) and Lemma 4.4, for all the points $\{\mathbf{x}^{(i)}\},\ i = 0, \cdots, u$ on the strict path, we have

$$H(\mathbf{x}^{(i)};\mathbf{x}^*;\alpha,\beta) < H(\mathbf{x}^{(i-1)};\mathbf{x}^*;\alpha,\beta) \leq -1/a, i = 1, \cdots, u,$$

when β satisfies (4.2). Especially,

$$H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta) < H(\mathbf{x}^{(u-1)};\mathbf{x}^*;\alpha,\beta) < H(\mathbf{x}^{(u-j)};\mathbf{x}^*;\alpha,\beta), j = 2, \cdots, u \tag{4.11}$$

On the other hand, since $\overline{\mathbf{x}} \in B^{**}$, $\overline{\mathbf{x}}+\overline{\mathbf{d}} \in B^{**}$ and $\overline{\mathbf{d}}$ satisfies (4.7), it follows from the definition of the function H that

$$\begin{aligned}
&H(\overline{\mathbf{x}}+\overline{\mathbf{d}};\mathbf{x}^*;\alpha,\beta) - H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta)\\
&= \tfrac{-\alpha}{a+f(\mathbf{x}^*)-f(\overline{\mathbf{x}}+\overline{\mathbf{d}})} + \tfrac{1}{a+f(\overline{\mathbf{x}})-f(\mathbf{x}^*)} + \tfrac{\alpha-1}{a} - 2\beta\\
&= \tfrac{[a+f(\mathbf{x}^*)-f(\overline{\mathbf{x}}+\overline{\mathbf{d}})]-\alpha[a+f(\overline{\mathbf{x}})-f(\mathbf{x}^*)]}{[a+f(\mathbf{x}^*)-f(\overline{\mathbf{x}}+\overline{\mathbf{d}})][a+f(\overline{\mathbf{x}})-f(\mathbf{x}^*)]} + \tfrac{\alpha-1}{a} - 2\beta\\
&= \tfrac{\alpha M_2[f(\mathbf{x}^*)-f(\overline{\mathbf{x}}+\overline{\mathbf{d}})]-M_1[f(\overline{\mathbf{x}})-f(\mathbf{x}^*)]-2a\beta M_1M_2}{aM_1M_2}.
\end{aligned} \tag{4.12}$$

It can be verified that the right hand side of (4.12) is strict positive when α satisfies (4.9) and β satisfies (4.2), that is

$$H(\overline{\mathbf{x}};\mathbf{x}^*;\alpha,\beta) < H(\overline{\mathbf{x}}+\overline{\mathbf{d}};\mathbf{x}^*;\alpha,\beta). \tag{4.13}$$

Therefore, $\overline{\mathbf{x}}$ minimizes $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ on the path $\{\mathbf{x}^{(i)}\}_{i=0}^{u+1}$ in $\mathbb{S}$ between $\mathbf{x}^*$ and $\mathbf{x}^{(u+1)} = \overline{\mathbf{x}}+\overline{\mathbf{d}}$. If $\overline{\mathbf{x}}+\overline{\mathbf{d}} = \mathbf{x}^{**}$, then the conclusion of the theorem holds. If $\overline{\mathbf{x}}+\overline{\mathbf{d}} \neq \mathbf{x}^{**}$, the path can be extended by the steepest descent trajectory $\{\mathbf{x}^{(i)}\}_{i=u+1}^{u+v}$ of f in $\mathbb{S}$ from $\overline{\mathbf{x}}+\overline{\mathbf{d}}$ to $\mathbf{x}^{**} = \mathbf{x}^{(u+v)}$, where $u+v=K$. Hence $\overline{\mathbf{x}}$ is a minimizer of $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ on the path $\{\mathbf{x}^{(i)}\}_{i=-1}^{u+v}$.□

Theorem 4.6 indicates that the function $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ satisfies the third condition of definition 7 when β satisfies (4.2) and α satisfying (4.9). From the analysis above, we can conclude that when β and α satisfy (4.2) and (4.9), $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ is a discrete filled function of the function f at the local minimizer $\mathbf{x}^*$ on $\mathbb{S}$.

5 Estimation of parameters α, β

The parameter values play important role in general filled function algorithms for global optimization, and generally are adjustable. A typical approach to select the parameter values in literature is to first give initial estimations of parameters, and then adjust the values of the parameters step by step in the process of implementing an algorithm to ensure the desired properties of filled functions. However, analyzing the characteristic of max-cut problems indicates that the values of the parameters α and β in the filled function H is independent of the variable $\mathbf{x} \in \mathbb{S}$ and need not to be adjusted in the implementation of the algorithm. It follows from Theorems 4.3 and 4.6 that when β and α satisfy (4.2) and (4.9), $H(\mathbf{x};\mathbf{x}^*;\alpha,\beta)$ is a desired filled function of f at $\mathbf{x}^*$ on $\mathbb{S}$, where $\mathbf{x}^*$ is a local minimizer of function f.

By (4.2), the value $(4qM+l_0)/(2a(a+4qM+l_0))$ increases and converges to $1/(2a)$ as $(4qM+l_0)$ increases and tends to ∞. Clearly, when $\beta = 1/(2a)$,

$$\beta = \frac{1}{2a} > \frac{4qM+l_0}{2a^2+2a(4qM+l_0)}$$

holds for all max-cut problems. In the rest of this section, we discuss the estimation for the value of the parameter α.

Since $0 < f(\overline{\mathbf{x}}) - f(\overline{\mathbf{x}}+\overline{\mathbf{d}}) \leq 4qM + l_0$, from inequality (4.7) and equation (4.10), we have

$$a + f(\overline{\mathbf{x}}) - f(\overline{\mathbf{x}}+\overline{\mathbf{d}}) - f(\mathbf{x}^*) \leq a + 4qM + l_0 - f(\mathbf{x}^*).$$

and

$$0 < M_2 \leq 2a + 4qM + l_0 - M_1. \tag{5.1}$$

Replacing $f(\overline{\mathbf{x}})-f(\mathbf{x}^*)$ and $f(\mathbf{x}^*)-f(\overline{\mathbf{x}}+\overline{\mathbf{d}})$ in (4.9) with M_2-a and M_1-a, respectively, and using (5.1) generate

$$\begin{aligned} \frac{2a\beta M_1M_2+M_1[f(\overline{\mathbf{x}})-f(\mathbf{x}^*)]}{M_2[f(\mathbf{x}^*)-f(\overline{\mathbf{x}}+\overline{\mathbf{d}})]} &= \frac{(2a\beta+1)M_1M_2-aM_1}{M_1M_2-aM_2} = \frac{2a\beta+1-\frac{a}{M_2}}{1-\frac{a}{M_1}} \\ &\leq \frac{2a\beta+1-\frac{a}{2a+4qM+l_0-M_1}}{1-\frac{a}{M_1}}. \end{aligned} \tag{5.2}$$

Let

$$\mu(t)=\frac{2a\beta+1-\frac{a}{2a+4qM+l_0-t}}{1-\frac{a}{t}}, a<t\leq a+4qM+l_0.$$

It can be verified that $\mu(t)$ is a monotonically decreasing function with respect to t in the interval $(a, a+4qM+l_0]$.

Let ϵ $(0<\epsilon\leq 1)$ denote the precision of the entries of the matrix L, then

$$M_1=a+f(\mathbf{x}^*)-f(\overline{\mathbf{x}}+\overline{\mathbf{d}})\geq a+\epsilon>a>0. \tag{5.3}$$

Hence, from (5.2), (5.3) and $\beta=1/(2a)$,

$$\begin{aligned} \frac{2a\beta+1-\frac{a}{M_2}}{1-\frac{a}{M_1}}\leq\mu(M_1)\leq\mu(a+\epsilon) &= \frac{2a\beta+1-\frac{a}{2a+4qM+l_0-(a+\epsilon)}}{1-\frac{a}{a+\epsilon}} \\ &= \frac{2(a+\epsilon)-\frac{a(a+\epsilon)}{4qM+l_0+(a-\epsilon)}}{\epsilon} \end{aligned} \tag{5.4}$$

The value of the right hand side in (5.4) increases and converges to $2(a+\epsilon)/\epsilon$ as the value $(4qM+l_0)$ increases and tends to ∞. Especially, if $\alpha=2(a+\epsilon)/\epsilon$, we have

$$\alpha=\frac{2(a+\epsilon)}{\epsilon}>\frac{2(a+\epsilon)-\frac{a(a+\epsilon)}{4qM+l_0+(a-\epsilon)}}{\epsilon},$$

and inequality (4.9) holds for all max-cut problems when the precision of the entries in the matrix L is within ϵ.

6 The Algorithm

In this section, we describe the discrete filled function algorithm embedded with the continuous relaxation algorithm (CA) for the solution of max-cut problems. The steps of the proposed algorithm and the details of implementing the algorithm will be presented in the following subsections.

6.1 Minimizing f Using Algorithm 1-NLS

Let $\mathbf{x} = (x_1, \cdots, x_n)^T \in \mathbb{S}$ be a feasible point of problem (MMC), the 1-neighborhood search (1-NLS) staring from the point $\mathbf{x}$ will be implemented to either find a local minimizer of function $f(\mathbf{x})$ on $\mathbb{S}$, or confirm that $\mathbf{x}$ already is a local minimizer. Denote the n neighbor points of $\mathbf{x}$ in $N(\mathbf{x})$ by $\mathbf{x}^1, \cdots, \mathbf{x}^p, \cdots, \mathbf{x}^n$, where the superscript p means that only the p^{th} element of $\mathbf{x}^p$ differs from the p^{th} element of $\mathbf{x}$. The point $\mathbf{x}^{i_*} \in N(\mathbf{x})$ satisfying $f(\mathbf{x}^{i_*}) = \min\{f(\mathbf{x}^p) : \ p = 1, \cdots, n\}$ is determined. If $f(\mathbf{x}^{i_*}) < f(\mathbf{x})$, then replace $\mathbf{x}$ with $\mathbf{x}^{i_*}$ and repeat the process until a point $\mathbf{x}^*$ satisfying $f(\mathbf{x}^*) = \min\{f(\mathbf{x}) : \mathbf{x} \in N(\mathbf{x}^*)\}$ is found. Formally, the 1-neighborhood local search (1-NLS) for problem (MMC) can be presented as follows.

Algorithm (1-NLS):

Step 1. Input a feasible point $\mathbf{x}_0 \in \mathbb{S}$, $f_0 = f(\mathbf{x}_0)$ and set $k = 0$.

Step 2. Calculate $f^i = f(\mathbf{x}_k^i)$ for all $i = 1, \cdots, n$ and $f^{i_*} = f(\mathbf{x}_k^{i_*}) := \min_i\{f^i\}$.

Step 3. If $f^{i_*} \geq f_k = f(\mathbf{x}_k)$, then return $\mathbf{x}_k$ as a local minimizer of f and stop;
Otherwise set $\mathbf{x}_{k+1} = \mathbf{x}_k^{i_*}$, $f_{k+1} = f^{i_*}$, $k = k + 1$, goto Step 2.

6.2 The Statement of the Algorithm

Let $\mathbf{x}^{CA}$ be a local solution of the problem (MX) obtained by using the continuous relaxation algorithm (CA) from an initial point $\mathbf{x}_0 \in [-1, 1]^n$. Then $\mathbf{x}^{(1)}$ =sign($\mathbf{x}^{CA}$) is a feasible point of the problem (MMC). Since $\mathbf{x}^{(1)}$ may not be a local minimizer of problem (MMC), local searches from $\mathbf{x}^{(1)}$ are implemented to either obtain a local minimizer of problem (MMC) or confirm that $\mathbf{x}^{(1)}$ is already a local minimizer. We use $\mathbf{x}^*$ to denote the minimizer. After the local minimizer $\mathbf{x}^*$ is obtained, initial points are successively and randomly generated in the neighborhood $N(\mathbf{x}^*)$ to minimize the filled function $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$ until that either a point, $\mathbf{x}^{(2)}$ say, in a basin of f lower than the basin containing $\mathbf{x}^*$ is obtained, or no progress in the function value can be obtained. In the case of obtaining $\mathbf{x}^{(2)}$, the process will be repeated until a point, $\mathbf{x}^{(k)}$ say, is obtained at which no progress in the function value can be obtained. Then $\mathbf{x}^{(k)}$ will be accepted as an approximate solution of problem (MMC). We denote the algorithm that generates a local minimizer of (MMC) using Algorithm CA + 1-NLS by discrete filled function embedded with continuous relaxation (**DFFEC**) algorithm. The algorithm (**DFFEC**) can be stated as follows.

Algorithm (DFFEC).

Step 0. (Initialization)

(1). Input the matrix W, the value of ϵ, constant a, an initial point $\mathbf{x}_0 \in [-1,1]^n$, an integer $n_1 (\leq n)$ and set $k := 0$;

(2). Calculate $\beta = 1/2a,\ \alpha = 2(a+\epsilon)/\epsilon$;

Step 1. Find a local minimizer $\mathbf{x}_k^{CA}$ of problem (MX) from $\mathbf{x}_k$ using **Algorithm CA**, and set $\bar{\mathbf{x}}_k^{CA}$ =sign($\mathbf{x}_k^{CA}$);

Step 2. Find a local minimizer $\mathbf{x}_k^*$ of problem (MMC) from $\bar{\mathbf{x}}_k^{CA}$ using **Algorithm 1-NLS**. Set $I = 1$, and $\widetilde{N} = \{\mathbf{x}_k^*\}$;

Step 3. Randomly generate an initial point $\mathbf{x}_k^i$ in $N(\mathbf{x}_k^*) \setminus \widetilde{N}$;

Step 4. Minimize the filled function $H(\mathbf{x}; \mathbf{x}_k^*; \alpha, \beta)$ starting from the point $\mathbf{x}_k^i$. If a point $\overline{\mathbf{x}}$ is obtained such that either $\overline{\mathbf{x}}$ satisfies $f(\overline{\mathbf{x}}) < f(\mathbf{x}_k^*)$ or $\overline{\mathbf{x}}$ lies in a basin of f lower than the current basin, then set $\mathbf{x}_{k+1} = \overline{\mathbf{x}}$, $k = k+1$ and goto Step 1;

Step 5. If $I < n_1$, set $\widetilde{N} = \widetilde{N} \bigcup \{\mathbf{x}_k^i\}$, $I = I+1$, and goto Step 3; Otherwise, return $\mathbf{x}_k^*$ and stop.

Remark 1. The index i of $\mathbf{x}_k^i$ in step 3 means that only the i^{th} element of $\mathbf{x}_k^i$ differs from that of $\mathbf{x}_k^*$. Note that the index i is distinct from but dependent on the counter I (see subsection 6.3 below).

Remark 2. The termination condition $I = n_1$ indicates that we can not get a better solution than the current $\mathbf{x}_k^*$ after randomly n_1 points out of n points in $N(\mathbf{x}_k^*)$ have been used as initial points to minimize the filled function $H(\mathbf{x}; \mathbf{x}_k^*; \alpha, \beta)$.

We can also use only the 1-NLS algorithm in subsection 6.1 to generate a local minimizer of problem (MMC) from a given feasible point. Then the minimization of the filled function can be implemented from the neighbor points of the local minimizer. We denote this algorithm that obtains a local minimizer of f using only 1-NLS by pure discrete filled function (**PDFF**) algorithm. The steps of the algorithm (**PDFF** are described as follows.

Algorithm (PDFF).

Step 0. (Initialization)

Given an initial $\mathbf{x}_0 \in [-1,1]^n$, other parameters are the same as **Algorithm DFFEC**.

Step 1. Find a local minimizer $\mathbf{x}_k^*$ of problem (MMC) from $\mathbf{x}_k$ using **1-NLS**, Set $I = 1$, and $\widetilde{N} = \{\mathbf{x}_k^*\}$.

The rest steps of Algorithm PDFF are the same as Step 3 to Step 5 of Algorithm DFFEC.

6.3 Generating Initial Points to Minimize H

Let $\mathbf{x}_k^*$ be a local minimizer of function $f(\mathbf{x})$ in $[-1,1]^n$. Then the minimization of the filled function H will be implemented from the neighbor points of $\mathbf{x}_k^*$ in $N(\mathbf{x}_k^*)$. These points are randomly generated in $N(\mathbf{x}_k^*)$. Now, we state the process of randomly generating $n_1(< n)$ (in case necessary) initial points from n points in $N(\mathbf{x}_k^*) \setminus \{\mathbf{x}_k^*\}$. Let $I_1 = [n/n_1]$, where $[a]$ denotes the integer part of the real number a. If n/n_1 is an integer, then we can partition the set $N = \{1, \cdots, n\}$ into n_1 disjoint subsets, and each subset has I_1 integers, that is, $N = N_1 \bigcup N_2 \bigcup \cdots \bigcup N_{n_1}$ and $N_1 = \{1, \cdots, I_1\}, N_2 = \{I_1+1, \cdots, 2I_1\}, \cdots, N_{n_1} = \{(n_1-1)I_1+1, \cdots, n_1 I_1\}$. Then $\mathbf{x}_k^i$ is selected as an initial point to minimize the filled function $H(\mathbf{x}; \mathbf{x}^*; \alpha, \beta)$, where $i = (I-1)\cdot I_1 + [c \cdot I_1] + 1 \in N_I, I = 1, 2, \cdots, n_1$ and c is a random number generated in $(0,1)$ and may be different for different I. Note that the superscript i in $\mathbf{x}_k^i$ means that only the ith element of $\mathbf{x}_k^i$ differs from that of $\mathbf{x}_k^*$.

If n/n_1 is a fraction, then set $n_0 = n - n_1 I_1$ $(n_0 < n_1)$. The set $N = \{1, \cdots, n\}$ is also partitioned into n_1 disjoint subsets, where each subset of the first $n_1 - n_0$ subsets has I_1 integers and each subset of last n_0 subsets has $I_1 + 1$ integers. Then the superscript i in initial point $\mathbf{x}_k^i$ is determined by

$$i = (I-1)\cdot I_1 + [c \cdot I_1] + 1;$$

for $I = 1, 2, \cdots, n_1 - n_0$, or

$$\begin{aligned} i &= (n_1 - n_0)I_1 + [(I_1+1)\cdot c] + 1 + (I_1+1)(I - (n_1 - n_0) - 1) \\ &= [(I_1+1)\cdot c] + 1 + (I_1+1)(I-1) - (n_1 - n_0), \end{aligned}$$

for $I = n_1 - n_0 + 1, \cdots, n_1 - n_0 + j, \cdots, n_1$, where $c \in (0,1)$ is still a random number generated in $(0,1)$.

6.4 Minimizing the Filled Function H

In this subsection, we describe the process of minimizing the filled function H. Let $\mathbf{x}_k^* = (x_{k1}^*, \cdots, x_{km}^*, \cdots, x_{kn}^*)^T \in \mathbb{S}$ be the current local minimizer of problem (MMC) and $\overline{\mathbf{x}} = (\overline{x}_1, \cdots, \overline{x}_n)^T \in \partial N(\mathbf{x}_k^*, K), 1 \le K \le n$. Denote $\mathcal{X}_K = \{\mathbf{y} \in \mathbb{S} : \mathbf{y} \in N(\overline{\mathbf{x}}), \|\mathbf{y} - \mathbf{x}_k^*\|_1 > \|\overline{\mathbf{x}} - \mathbf{x}_k^*\|_1\}$, then the following result is clear and its proofs is omitted.

Lemma 6.1 *There exist* $(n-K)$ *elements in the set* $\mathcal{X}_K$. □

Let $\mathbf{x}_k^i = (x_{k1}^i, \cdots, x_{km}^i, \cdots, x_{kn}^i)^T$ be a point that is randomly generated in $N(\mathbf{x}_k^*)\backslash\{\mathbf{x}_k^*\}$ using the method given in subsection 6.3. Since $\mathbf{x}_k^i \in \partial N(\mathbf{x}_k^*, 1)$, $|\mathcal{X}_1| = n-1$, where $\mathcal{X}_1 = \{\mathbf{x} \in \mathbb{S} : \mathbf{x} \in N(\mathbf{x}_k^i), \|\mathbf{x} - \mathbf{x}_k^*\|_1 > \|\mathbf{x}_k^i - \mathbf{x}_k^*\|_1\}$, and

$$\mathcal{X}_1 = \{\mathbf{x}_k^{i1}, \cdots, \mathbf{x}_k^{i,i-1}, \mathbf{x}_k^{i,i+1}, \cdots, \mathbf{x}_k^{ij}, \cdots, \mathbf{x}_k^{in}\},$$

$\mathbf{x}_k^{ij} = (x_{k1}^{ij}, \cdots, x_{km}^{ij}, \cdots, x_{kn}^{ij})^T \in \mathcal{X}_1$ with $x_{km}^{ij} = x_{km}^i, (m = 1, 2, \cdots, n, m \neq j)$, and $x_{kj}^{ij} = -x_{kj}^i$.

Now, we present the process of minimizing the filled function H starting from the initial point $\mathbf{x}_k^i \in N(\mathbf{x}_k^*)$. Assume that $f(\mathbf{x}_k^{ij}) \geq f(\mathbf{x}_k^*)$ holds for all points $\mathbf{x}_k^{ij} \in \mathcal{X}_1, (j = 1, \cdots, n, j \neq i)$ (If there exists $\mathbf{x}_k^{ij} \in \mathcal{X}_1$ such that $f(\mathbf{x}_k^{ij}) < f(\mathbf{x}_k^*)$, then $\mathbf{x}_k^{ij}$ is in a basin lower than the current basin containing $\mathbf{x}_k^*$). Since $f(\mathbf{x}_k^i) \geq f(\mathbf{x}_k^*)$, $\|\mathbf{x}_k^{ij} - \mathbf{x}_k^*\|_1 > |\mathbf{x}_k^i - \mathbf{x}_k^*\|_1$, and α, β are given in algorithm (DFFEC), it follows from Lemma 4.4 that

$$\delta_H(j) = H(\mathbf{x}_k^{ij}; \mathbf{x}_k^*; \alpha, \beta) - H(\mathbf{x}_k^i; \mathbf{x}_k^*; \alpha, \beta) < 0, j = 1, \cdots, n, j \neq i.$$

On the other hand,

$$\begin{aligned} \delta_H(j) &= \frac{f(\mathbf{x}_k^{ij}) - f(\mathbf{x}_k^i)}{[a + f(\mathbf{x}_k^{ij}) - f(\mathbf{x}^*)][a + f(\mathbf{x}_k^i) - f(\mathbf{x}^*)]} - 2\beta \\ &= \frac{\delta_f(j)}{[\delta_f(j) + a + f(\mathbf{x}_k^i) - f(\mathbf{x}^*)][a + f(\mathbf{x}_k^i) - f(\mathbf{x}^*)]} - 2\beta, \ j = 1, \cdots, n, \ j \neq i. \end{aligned}$$

where $\delta_f(j) = f(\mathbf{x}_k^{ij}) - f(\mathbf{x}_k^i)$. It can be verified that, for the fixed point $\mathbf{x}_k^i$, $\delta_H(j)$ is monotonically increasing with respect to the value of $\delta_f(j)$. Let

$$j_* = \arg\min_j \{\delta_f(j) : j \in \{1, 2, \cdots, n\} \setminus \{i\}\}$$

then for all $\mathbf{x}_k^{ij} \in \mathcal{X}_1$, we have

$$H(\mathbf{x}_k^{ij_*}; \mathbf{x}_k^*; \alpha, \beta) \leq H(\mathbf{x}_k^{ij}; \mathbf{x}_k^*; \alpha, \beta),$$

and hence $\mathbf{x}_k^{ij_*}$ is accepted as the next iterate point for minimizing $H(\mathbf{x}; \mathbf{x}_k^*; \alpha, \beta)$.

After $\mathbf{x}_k^{ij_*} \in \partial N(\mathbf{x}_k^*, 2)$ is found, the process above will be repeated in the set $\mathcal{X}_2 = \{\mathbf{x} \in \mathbb{S} : \mathbf{x} \in N(\mathbf{x}_k^{ij_*}), \|\mathbf{x} - \mathbf{x}_k^*\|_1 > \|\mathbf{x}_k^{ij_*} - \mathbf{x}_k^*\|_1\}$ with $|\mathcal{X}_2| = n-2$ until either a point in a lower basin is found or the minimization of H from the point $\mathbf{x}_k^i$ is terminated when K reaches $[n/2]$. Formally, the minimization of filled function H can be described as follows.

Algorithm (MF): Minimizing the Filled Function H

Step 1. Let $\mathbf{x}_k^i \in N(\mathbf{x}_k^*) \setminus \widetilde{N}$ be a randomly generated point, set $\widetilde{\mathbf{x}}_k^i = \mathbf{x}_k^i$. Calculate

$$\delta_0(j_*) = f(\widetilde{\mathbf{x}}_k^i) - f(\mathbf{x}_k^*) = -4 \sum_{m=1, m\neq i}^{n} x_{km}^* l_{mi} x_{ki}^*,$$

set $K = 1$ and $\widetilde{I} = \{i\}$.

Step 2. If $K = [\frac{n}{2}]$, go to step 2 of Algorithm (DFFEC). Otherwise, calculate

$$\begin{aligned} \delta_f(j) &= f(\mathbf{x}_k^{ij}) - f(\widetilde{\mathbf{x}}_k^i) = -4 \sum_{m=1, m\neq j}^{n} \widetilde{x}_{km}^i l_{mj} \widetilde{x}_{kj}^i \\ &= -4\widetilde{x}_{kj}^i (\widetilde{\mathbf{x}})^T \mathbf{l}_j + 4 l_{jj}, \ j = 1, \cdots, n, \ j \notin \widetilde{I}, \end{aligned} \tag{6.1}$$

for all $\mathbf{x}_k^{ij} \in \mathcal{X}_K = \{\mathbf{x} \in N(\widetilde{\mathbf{x}}_k^i) : \|\mathbf{x} - \mathbf{x}_k^*\|_1 > \|\widetilde{\mathbf{x}}_k^i - \mathbf{x}_k^*\|_1\}$, and

$$j_* = \arg\min\{\delta_f(j) : j \in \{1, 2, \cdots, n\} \setminus \widetilde{I}\}, \tag{6.2}$$

$$\delta_K(j_*) = f(\mathbf{x}_k^{ij_*}) - f(\mathbf{x}_k^*) = \delta_{K-1}(j_*) + \delta_f(j_*),$$

Step 3. If $\delta_K(j_*) = f(\mathbf{x}_k^{ij_*}) - f(\mathbf{x}_k^*) < 0$, then go to Step 1 in algorithm (DFFEC) to find a better local minimizer of problem (MX) starting from $\mathbf{x}_k^{ij_*}$.

Step 4. Set $\widetilde{\mathbf{x}}_k^i = \mathbf{x}_k^{ij_*}$, $K = K + 1, \widetilde{I} = \widetilde{I} \cup \{j_*\}$, goto step 2.

Remark 3. By the symmetrical structure of the set $\mathbb{S}$, if $\mathbf{x} \in \partial N(\mathbf{x}_k^*, K)$ for any integer K with $[\frac{n}{2}] < K \leq n$, then $-\mathbf{x} \in \partial N(\mathbf{x}_k^*, n - K)$ and $f(-\mathbf{x}) = f(\mathbf{x})$. Thus if we can not find a point $\overline{\mathbf{x}}$, that will be in a basin of f lower than the current basin, along the steepest descent path of $H(\mathbf{x}; \mathbf{x}_k^*; \alpha, \beta)$ from the point $\mathbf{x}_k^i \in N(\mathbf{x}_k^*)$ to a point in $\partial N(\mathbf{x}_k^*, [\frac{n}{2}])$, then it is not necessary to continue the search in $\partial N(\mathbf{x}_k^*, [\frac{n}{2}] + 1)$. Hence, when K reaches $[\frac{n}{2}]$ in step 2, the minimization of filled function $H(\mathbf{x}; \mathbf{x}_k^*; \alpha, \beta)$ need to be restarted by generating another initial point in $N(\mathbf{x}_k^*)$.

Remark 4. In order to obtain the next iterate point $\mathbf{x}_k^{ij_*}$ from previous point $\widetilde{\mathbf{x}}_k^i$ in minimizing H, instead of directly calculating the value $f(\mathbf{x}_k^{ij})$ ($j \in \{1, 2, \cdots, n\} \setminus \widetilde{I}$), inner products of two vectors in (6.1) (at most $(n - |\tilde{I}|$ inner products) need to be calculated.

7 Numerical Results

In this section, some numerical results and comparisons are reported on some typical max-cut test problems. The proposed algorithm in this paper is coded using Matlab 6.5 and is implemented in a PC with 1.43GHz Pentium IV and 256Mb of RAM. The following notations are used in this section.

DF-x: The best max-cut value obtained by running Algorithm DFFEC *x* times. For instance, *DF-1* means that the max-cut value is obtained by running Algorithm DFFEC once.

PDF-x: The best max-cut value obtained by running Algorithm PDFF *x* times.

VNS-x: The best max-cut value obtained by running VNS heuristic *x* times.

Cn-m: The best max-cut value obtained by running CirCut with parameters *(n,m)*.

UB: An upper bound of the max-cut value obtained by the SDP relaxation.

ρ-x: The ratio of *DF-x* and *UB* when running Algorithm DFFEC *x* times.

T-x: The average time of CPU (in seconds) in running certain algorithm *x* times.

N-x: The average numbers of local minimizers of problem (MC) returned by running Algorithm DFFEC *x* times;

NH-x: The average times of minimizing the filled function H when running Algorithm DFFEC *x* times.

All the entries w_{ij} in the matrix W for all max-cut test problems are integers, and hence the precision of entries is $\epsilon = 1$, and parameter values $a = 0.5$, $\beta = 1$, $\alpha = 3$ satisfy the conditions when running Algorithm DFFEC (see Step 0 in algorithm DFFEC).

The first set of test problems consists of 45 max-cut problems. These problems, denoted by $Pi, (i = 1, \cdots, 45)$, are randomly generated by the following way. Let $u \in (0, 1)$ be a given constant and i, j be two nodes. Then generating a random fraction $v \in (0, 1)$. If $v \leq u$, then there is an edge between the nodes i and j, and the weight is $w_{ij} = 1$ for $n = 100$ and $n = 150$, and w_{ij} is a random integer between 1 and 50 for $n = 200$. If $v > u$, then $w_{ij} = 0$, that is, there is no edge between nodes i and j. The second set of test problems consists of 15 *G-set* graphs, G1, G2, G3, G11, G12, G13, G14, G15, G16, G22, G23, G24, G43, G44 and G45. These problems are created using a graph generator, **rudy**, written by Pro. Rinaldi.

Numerical comparisons by running both the Algorithm DFFEC and Algorithm PDFF once on test problems $Pi, (\ i = 1, \cdots, 9$ and $i = 31, \cdots, 45)$ are given in Table 1. It can be observed from Table 1 that PDFF can get almost the same solution as

DFFEC when the problem size and the value of u are small. However, DFFEC obviously generates better solutions than PDFF does when either the problem size or the value of u increases. Based on these numerical results, in the remainder of this section, we only give the comparison of the Algorithm DFFEC with other available approximation algorithms.

The first comparison is given between the Algorithm DFFEC and VNS heuristic for problems $Pi, (i = 1, \cdots, 45)$ and 10 out of 15 G-set problems. Variable Neighborhood Search (VNS) heuristic is a local search algorithm and is one of the algorithms considered by Festa et al. [6] in finding the solution of max-cut problems. The numerical results in [6] show that VNS heuristic is one of promising algorithms with high performance. For all problems $Pi(i = 1, \cdots, 45)$, Both the Algorithm (DFFEC) with $n_1 = [n/2]$ and the VNS heuristic with $k_{max} = [n/2]$ (see [6], [1], [2]) are running 5 times for each problem of the first set test problems (see Table 2) and one time for each problem of the second test problems (see Table 3). The results of VNS heuristic on the 10 G-set problems are completely quoted from Table I in [6]. It can be observed from Table 2 and Table 3 that Algorithm DFFEC generates better max-cut values than VNS heuristic for most of these test problems with only the exception of G2.

The Second comparison is given between the algorithm (DFFEC) and the CirCut heuristic. Based on the Goemans-Williamson randomized algorithm, Burer et al. [4] proposed a rank-two relaxation for max-cut problems and developed a specialized version of the Goemans-Williamson technique. Burer et al. [4] implemented their approach by a Fortran 90 code named "CirCut". It is one of the most popular heuristics for solving max-cut problems. In this comparison, the algorithm (DFFEC) is running one time and 10 times, respectively, on 8 G-set problems, G11, G12, G13, G14, G15, G22, G23, and G24. The algorithm CirCut Heuristic is running with parameter values *(n,m)=(0,1)* and *(n,m)=(5,10)*, respectively. The results are given in Table 4, where results of *Cn-m* is completely cited from Burer et al. [4]. It can be seen from Table 4 that the results of *DF-1* are better than those of *C0-1* with the exception of G23. When the parameter values *(n,m)=(5,10)* are used in running CirCut heuristic, only the results of problems G22, G23 are better than those of *DF-10*.

Finally, we report results on third class of test problems, the benchmark problem set of the 7th DIMACS Implementation Challenge (see http://dimacs. rutgers.edu/ Challenges/ Seventh/). Two problems, pm3-8-50 and g3-8 are chosen. These two problems

Table 1: The Comparisons of the value *DF-1* and *PDF-1* for test problems P1 - P9, and P31 - P45, where *DF-1* and *PDF-1* mean the objective value obtained by running DFFEC and PDFF only one time, respectively.

			Max-cut value		*Time*	
Name	n	u	*DF-1*	*PDF-1*	*DF-1*	*PDF-1*
P1	100	0.1	138	138	6.21	12.32
P2			128	126	6.08	12.01
P3			130	130	6.32	11.52
P4	100	0.3	294	292	5.81	11.86
P5			323	321	5.43	10.97
P6			335	330	5.60	11.05
P7	100	0.5	455	452	4.87	10.16
P8			500	500	5.91	9.79
P9			469	465	8.10	15.21
P31	200	0.1	12684	12676	96.35	137.39
P32			13135	13123	96.76	112.63
P33			12370	12354	96.52	134.46
P34	200	0.3	30842	30832	115.67	141.23
P35			30589	30573	71.78	132.19
P36			29702	29670	73.53	125.36
P37	200	0.5	43830	43812	61.98	130.32
P38			44809	44795	81.63	101.42
P39			45323	45311	68.62	92.13
P40	200	0.7	58496	58482	95.79	103.20
P41			60085	60070	92.51	101.87
P42			57444	57430	61.86	101.36
P43	200	0.9	69145	69131	95.13	115.78
P44			68489	68473	105.61	124.79
P45			67481	67469	95.19	112.47

were generated by Jünger and Liers using the Ising model of spin glasses. The sizes of both the problems are the same with 512 nodes and 1536 edges. Experiments show that the algorithm CirCut needs to run 100 times to generate the objective values 452 for pm-3-8-50 and 4.13946e+7 for g3-8, while the Algorithm DFFEC only needs to run 10 times to generate better objective values 456 for pm-3-8-50 and 4.16738e+7 for g3-8.

For large scale max-cut problems, the computational time of the proposed algorithm DFFEC is high. Table 5 gives the calculation costs of the proposed algorithm in the minimization of the problem (MC) and the minimization of filled functions, respectively. It can be seen from Table 5 that most of the calculation costs are in the minimization of filled functions, and that larger *NH-x* is, more CPU-time is. It is clear that *NH-x* increases as the preset integer n_1 and the problem dimension n.

The following observations can be obtained based on the results in Table 2, Table

3 and Table 4.

(1) The ratio ρ in Table 2 and % of UB in the last row of Table 3 indicate that the solutions obtained by the proposed algorithm DFFEC are very close to global solutions for all test problems and show that Algorithm DFFEC is efficient in finding satisfactory solutions of max-cut problems.

(2) Although we employ the continuous relaxation algorithm (CA) to find local minimizers of problem (MX) and avoid parameters adjusting to reduce the computation cost, the CPU-time is still expensive. This is because N, the number of the local minimizers of the function f, and NH, the times of minimizing the filled function H, are large. That is, larger N and NH are, more CPU-time is.

(3) It is important to note that the costs of running VNS and CirCut heuristic were obtained on a SGI Challenge computer (with 28 196-Mhz MIPS R10000 processors and 7.6 Gb of memory) [6] and a SGI Original 2000 machine [4], respectively, while our results in costs are obtained on a PC with 1.43GHz and 256Mb of RAM. It can be understood that the PC with 1.43GHz Pentium and 256Mb of RAM is not comparable with their machines.

8 Conclusions and remarks

Based on the continuous relaxation algorithm in [20] and the new filled function, a discrete filled function algorithm in finding approximation to global solutions of max-cut problems is proposed. Since the parameters of the new filled function are exactly the same for all max-cut problems that avoids the adjusting of parameters, and the skill of randomly generating initial points for the minimization of the new filled function is adopted, the efficiency of the proposed algorithm is greatly increased.

Further works on the proposed algorithm are required to refine the algorithm in theory and implementation. As mentioned by one reviewer, when the VNS heuristic is combined into the discrete filled function algorithm, whether the performance of solutions can be further improved or not. We hope that the answer is positive, that is, the numerical results generated by such an algorithn would be better than the current case, but still needs to be tested and verified.

Acknowledgments

The authors are indebted to three anonymous referees whose comments helped considerably to improve this paper. We thank the second referee to provide us the literatures [5, 15, 22] on discrete filled function methods.

References

[1] D. Abraham, S.Ángel, F.Felipe and C. Raúl, A Low-Level Hybridization between Memetic Algorithm and VNS for the Max-Cut Problem. (2005)999-1006.

[2] S.M. Antonio, D. Abraham, J.P.Juan and C.Raúl, High-Performance VNS for the Max-Cut Problem using Commodity Graphics Hardware. (2005), Manuscript.

[3] F.Barahona, Grötschel. and G. Reinelt,(1988), An application of combinotiorial optimization to statistical physics and circuit layout design. Oper.Res.36,493-513.

[4] S.Burer, R.D.C. Monteiro , and Y.Zhang, (2001), Rank-two Relaxation Heuristics for Max-cut and Other Binary Quadratic Programs. SIAM J.Opt. 12, 503-521.

[5] Z.-G. Feng and K.-L.Teo, A discrete filled function method for the design of FIR filters with signed-powers-of-two coefficients, IEEE Transactions on Signal Processing, Available On-Line and Accepted for Inclusion in a future Issue, 2007.

[6] P. Festa, P. M. Pardalos, M. G. C.Resende, and C. C. Ribeiro, (2002), Randomized Heuristics for the Max-Cut Problem, Optimization Methods and Software, 17:1033-1058.

[7] M. Garey, D. Johnson and L. Stochmeter, (1976), Some simplified NP-complete graph problems. Theoret. Comput. Sci. 1,237-267.

[8] R. P. Ge, (1990), A filled function method for finding a global minimizer of a function of several variables, Mathematical Programming 46, 191-204.

[9] R.P. Ge and C.B. Huang, (1989), A continuous approach to nonlinear integer programming, Appl. Math. Computat. 34 (1) 39-60.

[10] M.X. Goemans, D.P. Williamson, (1995), Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, J. Assoc. Comput. Mach. 42 (6) 1115-1145.

[11] C. Helmberg, F. Rendl, (2000), A spectral bundle method for semidefinite programming, SIAM J. Optim. 10 (3) 673-696.

[12] M. Jünger and F.Liers, private communication, Universität Köln, Cologne, Germany, 2000.

[13] R. M. Karp, (1972), Reducibility among combinatorial problems. In Complexity of Computer Computations, R. Miller and J. Thatcher, eds. Plenum Press, New York, pp85-103.

[14] X. Liu, (2006) A Class of Augmented Filled Functions, Computational Optimization and Applications, 33, 333-347,

[15] C.-K. Ng, D.Li and L.-S. Zhang,(2007) Discrete global descent method for discrete global optimization and nonlinear integer programming, Journal of Global optimization, 37, pp.357-379.

[16] Chi-kong Ng, L.-S. Zhang, D. Li and W.-W. Tian, (2005), Discrete Filled Function Method for Discrete Global Optimization, Computational Optimization and Applications, 31, 87-115.

[17] Y.-L. Shang, L.-S. Zhang, (2005), A filled function method for finding a global minimizer on global integer optimization, Journal of Computational and Applied Mathematics 181 200-210

[18] C.-X Xu, X.-L. He and F.-M. Xu, (2006), An Effective Continuous Algorithm for Approximate Solutions of Large Scale Max-Cut Problems. Journal of Computational Mathematics, Vol. 24(6), 749-760.

[19] Z. Xu, H.-X. Huang, Panos M. Pardalos and C.-X. Xu, (2001), Filled functions forunconstr ained global optimization, Journal of Global Optimization 20, 49-65.

[20] F.-M. Xu, C.X. Xu, H.-G. Xue, (2005), A Feasible Direction Algorithm Without Line Search for Solving Max-Bisection Problem, Journal of Computational Mathematics,23, 619-634.

[21] C.-X. Xu, J.-Z. Zhang, (2001), Survey of quasi-Newton equations and quasi-Newton methods for optimization, Annals of Operations Reseach, 103: 213-234.

[22] Y.-J. Yang and Y.-M. Ling, (2007) A new discrete filled function algorithm for discrete global optimization, Journal of Computational and Applied Math ematics, 202, No.2, pp.280-291.

[23] U. Zwick, (1999), Outward Rotations: A Tool for Rounding Solutions of Semidefinite Programming Relaxations, with Applications to Max cut and Other Problems, In Proc. of 31th STOC, pp. 679-687.

Table 2: The Numerical Comparisons of VNS Heuristic with Algorithm (DFFEC) on 45 random generated test problems at the same PC. The best solution found in running the two algorithms 5 times, respectively.

				VNS		DFFEC		
Name	n	u	*UB*	*VNS-5*	*T-5*	*DF-5*	*T-5*	ρ*-5*
P1	100	0.1	145	136	1.04	**140**	6.19	0.9655
P2			133	**128**	0.94	**128**	6.10	0.9624
P3			136	128	1.00	**131**	6.47	0.9632
P4	100	0.3	308	292	0.72	**294**	6.02	0.9545
P5			340	321	0.87	**323**	5.68	0.9500
P6			354	**337**	0.98	**337**	5.55	0.9520
P7	100	0.5	475	**455**	0.77	**455**	4.92	0.9579
P8			521	**500**	0.80	**500**	5.75	0.9597
P9			489	469	0.87	**471**	8.27	0.9632
P10	100	0.7	636	610	0.83	**612**	5.72	0.9623
P11			624	598	0.89	**599**	9.60	0.9599
P12			649	**624**	0.95	**624**	5.97	0.9615
P13	100	0.9	779	750	1.85	**752**	13.68	0.9653
P14			765	737	1.70	**740**	12.74	0.9673
P15			738	711	1.11	**712**	7.65	0.9648
P16	150	0.1	301	282	5.15	**284**	39.90	0.9435
P17			309	291	4.26	**295**	34.43	0.9547
P18			280	267	4.06	**271**	30.92	0.9679
P19	150	0.3	734	691	3.48	**697**	24.72	0.9496
P20			742	703	2.90	**705**	23.07	0.9501
P21			725	683	3.28	**690**	28.40	0.9517
P22	150	0.5	1071	1019	2.42	**1026**	26.29	0.9580
P23			1086	1032	2.75	**1040**	23.21	0.9576
P24			1091	1042	2.83	**1047**	23.60	0.9596
P25	150	0.7	1354	1304	2.50	**1308**	25.30	0.9660
P26			1380	1321	2.88	**1331**	30.67	0.9645
P27			1372	1318	2.68	**1322**	30.12	0.9636
P28	150	0.9	1633	1573	2.51	**1582**	29.03	0.9688
P29			1616	1562	2.46	**1567**	29.92	0.9697
P30			1651	1591	2.64	**1598**	25.92	0.9679
P31	200	0.1	13332	12624	15.59	**12687**	97.39	0.9516
P32			13871	13067	15.58	**13135**	94.56	0.9469
P33			13031	12286	17.15	**12372**	98.62	0.9494
P34	200	0.3	32423	30759	10.77	**30844**	121.77	0.9513
P35			32190	30466	7.87	**30593**	71.78	0.9504
P36			31165	29656	11.15	**29702**	73.48	0.9531
P37	200	0.5	45907	43527	6.57	**43834**	67.38	0.9548
P38			46840	44685	6.60	**44811**	86.19	0.9567
P39			47416	45282	9.00	**45323**	69.37	0.9559
P40	200	0.7	60660	58001	7.69	**58498**	94.48	0.9644
P41			62492	59873	7.48	**60089**	91.06	0.9615
P42			59790	56982	5.04	**57446**	68.75	0.9608
P43	200	0.9	71657	69014	7.05	**69149**	103.04	0.9650
P44			70911	**68489**	7.72	**68489**	113.72	0.9658
P45			69948	67443	7.77	**67485**	98.00	0.9649

Table 3: The Numerical Comparison with VNS Heuristic on 10 Helmberg and Rendl [11] *G-set* graph instances. Here, we only run Algorithm (DFFEC) one time. *UB*, and *VNS Heuristic* are completely borrowed from the TABLE I in Festa et al. [6]. The last two rows of this table list *sum of obtained max-cut objective value* and *sum of the optimal value of SDP relaxation* over the 10 instances, and the ratios of both sums

				VNS Heuristic		*DFFEC*	
Name	n	density	*UB*	*VNS-1*	*T-1*	*DF-1*	*T-1*
G1	800	6.12%	12078	11549	40.95	11570	993.30
G2			12084	11575	37.32	11571	922.12
G3			12077	11577	16.98	11577	977.83
G14	800	0.63%	3187	3040	12.89	3045	1125.64
G15			3169	3017	18.09	3032	1165.29
G16			3172	3017	10.30	3028	1032.37
G43	1000	2.10%	7027	6599	36.78	6607	1558.85
G44			7022	6559	40.55	6591	1504.36
G45			7020	6555	24.30	6574	1230.65
G22	2000	1.05%	14123	13087	56.98	13185	7582.29
		sum	80959	76575		76780	
		% of *UB*	100	94.58		94.84	

Table 4: The Numerical Comparison with CirCut code on 8 Helmberg and Rendl [11] G-set graph instances. Here, we run Algorithm (DFFEC) once and 5 times, respectively. The *CirCut* column cited from Burer et al. [4] using parameters *(n;m)* = (0; 1) and *(n;m)* = (5; 10).

				CirCut			*DFFEC*		
Name	n	density	*UB*	*C0-1*	*C5-10*	*T-10*	*DF-1*	*DF-10*	*T-10*
G11	800	0.63%	627	524	554	3.88	552	560	964.71
G12			621	512	552	3.76	550	554	1326.24
G13			645	536	572	3.45	572	578	1072.17
G14	800	1.58%	3187	3016	3053	5.53	3038	3055	1043.61
G15			3169	3011	3039	5.91	3027	3041	1146.13
G22	2000	1.05%	14123	13148	13331	22.31	13185	13329	3551.27
G23			14129	13197	13269	18.85	13181	13257	3269.09
G24			14131	13195	13287	27.30	13211	13313	3253.15

Table 5: The average number of Minimizing the Filled Function H and the average number of local minimizers of the function f obtained by the Algorithm (DFFEC) for part of test problems.

Name	*N-5*	*NH-5*	Name	*N-5*	*NH-5*	Name	*N-5*	*NH-5*
P16	10	142	P34	10	231	G1	12	161
P22	8	123	P37	10	163	G11	33	215
P28	12	127	P43	9	195	G14	32	190